Plan vs. Reality

After feedback, I did not change much in the grob architecture of classes, but there were a

lot of changes in methods and way of their communication.

Model: Classes for Map were planned correctly.

In AI I changed the way the Enemy Half Map was covered with MinimalCoveregeFinder class. In a way, that instead of searching for random tile on enemy's side, to calculate path there, MinimalCoverege algorithm was redesigned to search for nearest enemy tile instead of random one.

In GameLogic in order to implement CLI I pass PropertyChangeSupport. The biggest change was to implement State Maschine pattern to switch different Game Phases and adapt AI behavior accordingly.

Bug Fix: Implementation of good ErrorHandling and Logs helped to debug the project quickly and

effectively.

Converter: Converter was separated in 2 classes ServerMessageConverter and ClientMessageConverter. In order to improve Single Responsibility.

Controller: ClientManager class that implements Controller was not changed much. It implements the main logic and structure of the game.

Network: PlayerID should not be privately stored in Network class since it will be required in ClientManager to give to ServerMessageConverter, where convertFullMap and convertEMove will be invoked. In order to form PlayerHalfMap in ClientMessageConverter, UniquePlayerIdentifier is required.

View: This part changed the most archtitectually. Instead of implementation of PropertyChangeListener patten, I implemented one CLI class that uses java.beans.PropertyChangeEvent and java.beans.PropertyChangeListener to work with events.

Tests: Tests saved a lot of time during debugging phase as they checked the main AI logic. UnitTest cover all the classes except of Network and ClientManager, because they need avalid and actual gameID to perform correctly and are not supposed to be covered with UnitTests.